Security Concepts as Add-On for Process Models

Jacob Geisel, Brahim Hamid, Jean-Michel Bruel

Security Concepts as Add-On for Process Models

Jacob Geisel, Brahim Hamid, Jean-Michel Bruel
IRIT, University of Toulouse
Toulouse, France
{geisel,hamid,bruel}@irit.fr

***Abstract*—Development processes for software construction are common knowledge and widely used in most development organizations. Unfortunately, these processes often offer only little or no support in order to meet security requirements. In our work, we propose a methodology to enhance these process models with security concepts, backed by a security-oriented process model specification language. The methodology supports existing process models, which will be extended by established security approaches, as well as information security risk management standards, to fulfill the demand for secure software engineering. The methodology and the process modeling language we propose, have been successfully evaluated by the TERESA project for specifying development processes for trusted applications and integrating security concepts into existing process models.**

***Keywords*-Secure Software Engineering, Process Modeling, Repository, Reuse, Model-Driven Engineering**

I. Introduction

Development processes for software construction are common knowledge and mainstream practice in most development organizations. Unfortunately, these processes offer little support in order to meet security requirements and are rarely formalized. As a consequence, there are increased risks of security vulnerabilities that are introduced into software in various stages of development. Secure software (or software security) engineering aims to avoid security vulnerabilities in software by considering security aspects from the very beginning and throughout the life cycle. From another perspective, formalizing processes offers the ability to teach and communicate them and to reason about them.

The SEMCO[1] project tries to close this gap by offering a Model-Driven Engineering framework for, on the one hand, modeling and formalizing artifacts (e.g., security patterns) and on the other hand to provide methodologies for model-based development (e.g., pattern-based security-oriented engineering). Modeling is becoming a major paradigm in system engineering and particularly in system software engineering [1], but also in process engineering with the appearance of process metamodels [2].

In this work, we propose a process modeling environment, which associates model-driven paradigms and established security engineering concepts, to support the design of repository-centric security-oriented process models. In this context, we propose a methodology on enhancing existing process models through security aspects as well a metamodel to formalize development processes with security constraints. To enable reuse, common industry-relevant approaches for considering security aspects in process models are made available to process designers through a set of model libraries. As part of the assistance for the modeling of process models for secure applications, we implement a toolchain to support the different activities of process modeling and a repository, providing a set of reusable libraries. The proposed solutions were evaluated in the TERESA project[2].

The rest of this paper is organized as follows. Section II outlines existing work on process metamodels and (security-oriented) models. Section III shows our approach on adding security concepts to existing process models. Section IV outlines parts RCPM (Repository Centric Process Metamodel) for security-oriented process modeling. Section V describes how to use RCPM to formalize security-oriented processes and security type libraries. Section VI shows a case study from the Metrology Domain; a basic process model, formalized and augmented by a security type library. Section VII concludes this paper, discussing the advantages and limits of our approach and giving an outlook on future work.

II. Related Work

We will give an overview on the existing approaches on formalizing process models, on industry-relevant process models as well as on approaches on taking into account security concepts.

Process Metamodeling: Different process metamodels are proposed [3], [4] for modeling software engineering processes. These process metamodels are divided into different categories according to [5]. The viewpoint of process metamodel concentrates different aspects of methodologies that are used by these metamodels. In our context, process models will be created with the viewpoint of activity-oriented, as the development of security-oriented systems is more directly modeled in this viewpoint. SPEM2 (Software & Systems Process Engineering Metamodel) [3] was created by the OMG as a *de facto*, high-level standard for processes used in object-oriented software development. The scope of

[1] http://www.semcomdt.org

[2] http://www.teresa-project.org/

SPEM is purposely limited to the minimal elements necessary to define any software and system development process, without adding specific features for particular development domains or disciplines (e.g., project management, security). Other commonly used process metamodels like UMA or OPEN have similar characteristics.

Process Models: The V-Model [6] development process, also called verification & validation model, is suggested by the standard IEC61508. It is a trustworthy software development model, which aims at taming the complexity of project management, and which is used by big companies. The Rational Unified Process (RUP), an implementation of the Unified Process, is a comprehensive process framework that provides industry-tested practices for software engineering [4]. It is an iterative software development process framework, providing prototypes during each iteration.

Security Processes, Process Models and Process Standards: We outline the forefront representatives, as they are recognized as the major players in the field, as well as additional industry standards. Microsoft's Security Development Life cycle [7] is probably the most rigorous and more oriented towards large organizations, defining a process with guidance, allowing a management perspective to supervise the process. The Comprehensive, Lightweight Application Security Process [8] by the OWASP Consortium is a lightweight process, allowing customization to fit different projects and focussing on security as the central role of the system. McGraw's work [9] is based on industrial experience and has been validated over time, providing a set of best practices. Common Criteria for Information Technology Security Evaluation [10] is a ISO/IEC standard for computer security certification and defining a generic framework offering process designers to specify security functional requirements through protection profiles.

III. Approach

The methodology we propose is based on a repository of modeling artifacts. Once the repository is set up and populated with process model libraries and process type libraries, the (end-user) process engineer begins building domain specific process models. The central idea of our approach is to enable security-concepts in existing process models in a direct way. To achieve this, we map necessary concepts in *General Purpose Process Description Language (GPPDL)* to concepts of a *Security-Oriented Process Description Language (SOPDL)*, to be able to represent the existing process in a more adapted language. The second step is to add security concepts in a direct way and to be able to validate the process via existing rules. Once security concepts added and the process passes validation, a reverse mapping is done to produce a security enhanced process from the original one and the security oriented one.

The methodology is illustrated in Figure 1. On the left side is the workflow of the process in a *GPPDL* and on the right the actions taken on the process in a *SOPDL*. Steps **Step1** and **Step4** show the tasks of mapping a process from *GPPDL* to *SOPDL* and vice-versa. **Step1** does a *forward* mapping of the concepts used in the process model described in a *SOPDL* and allows the creation of a (semi-) complete process model in the *GPPDL*.

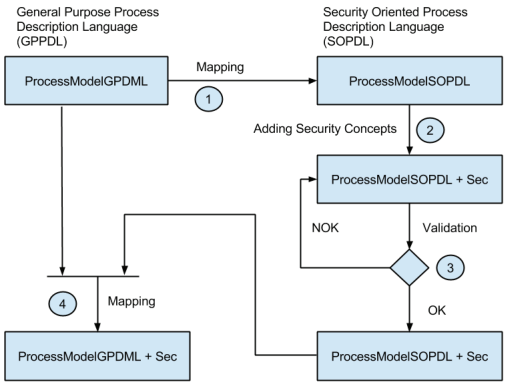

Figure 1. Add-On Methodology Overview

In the first step, **Step1**, optional if the process model is already defined in the *SOPDL*, allows process designers to represent their process models in the *SOPDL*. By mapping the concepts of the *GPPDL* to the concepts of *SOPDL* the designer can now use the process model in the supporting framework.

In the next step, **Step2**, the process designer is adding security concepts to the process in a simple and direct way using the *SOPDL*. In a first review on the process model, the designer can explicitly convert existing implicit concepts to explicit security concepts from the *SOPDL*. Then, existing knowledge can be added to the process model defined by type libraries, which are made available to the process designer via a repository of process type libraries.

Once the additional security concepts added to the process model, the process designer validates (**Step3**) the process model to several concerns. The first validation done is towards the *SOPDL*, to validate the conformance of the process model in terms of its syntax. The second validation allows the process designer to see if the integration of the model type library is complete and well done. If the validation in step **Step3** fails, action **Step2**, needs to be reiterated until the validation passes.

Several type libraries exist for different security approaches as well as for supporting engineering methods, such as Pattern-Based System Engineering (PBSE), and existing process models can be enhanced by multiple libraries. The last, optional, step (**Step4**), permits the process designer to

map the modified process model back and to merge it with the initial process model (*backwards* mapping). This allows the process designer to benefit of supplementary tooling available for the initial *PDL*.

IV. RCPM Concepts for Security and Reuse

The RCPM is a metamodel defining a new formalism for security process modeling based on a repository of modeling artifacts. The concepts of the metamodel[3], which are only briefly outlined have been presented in previous work [11], [12]. In the following we will outline the different packages. We will focus on sub-packages important for our and detail only these.

Core Package: The Core Package contains the elements which are used as top-level elements throughout the other packages and contain the basic attributes of all elements. These concepts include basic concepts (e.g., Element, Association) and their attributes (e.g., name, description).

Process Package: The Process Package contains all the concepts used for process engineering, the basic concepts, like Process Model, concepts of a work breakdown structure and concepts needed for detailing activities. This package is largely inspired by either existing process metamodels, such as SPEM2, UMA and/or OPF as well as by industry used process models such as the V-Modell XT.

Safety Engineering Package: Based on the Process Package, the Safety Engineering Package regroups recurring Safety Engineering Concepts and extends and enhances process concepts. The safety concepts of this package are derived from process models which are safety oriented, such as the V-Modell XT.

Security Engineering Package: The Security Engineering Package regroups recurring Security Concepts and Security Concepts linked to specific phases in the development life cycle, like Roles, Activities or Checkpoints (e.g., *SecurityEngineer*, *ThreatModeling*, *SecurityMetrics*, *CodeInspection*, *RequirementsAudit*, *RiskAssessment*). It is based on the Process and the Safety Package and reuses their concepts to express security-oriented concepts.

Type Package: The Types Package is used to define libraries for reuse of process blocks and to create constraints on Breakdown, Work Breakdown and Association Elements. The type elements correspond to the existing process elements and associations in the other packages (i.e., process, safety, security, repository). The Type Package allows to create reusable process blocks, using a structure similar to the existing (Work) Breakdown Structure and the Process/Pattern/Safety/Security Concepts, without the need to create a whole process model. These block are stored as process model libraries and can be loaded into existing process models. Elements in existing process models can now be typed by elements from the model library and are thus enhanced by the additional information.

[3]The complete description of the abstract syntax of the RCPM is available online http://www.semcomdt.org/semco/resources/RCPM.pdf

V. Security-Oriented Process Modeling

We propose an incremental specification process consisting of the following phases: (1) the specification of the security oriented process model, (2) the refinement using appropriate model libraries. The target representation is RCPM. The informal description given in Section IV reflects our understanding from the representation of security-oriented process modeling given in literature. To create model instances of the proposed metamodel, we provide a concrete syntax. We choose to use a EBNF grammar to define a concrete syntax for the RCPM language.

Model Transformation: The mappings expressed in the high-level description of the methodology are realized via model transformations in the context of process models formalized in a GPPDL and RCPM as SOPDL. The forward transformation extracts the *relevant* and *necessary* elements from the initial process and maps them to the corresponding elements in RCPM resulting in a (reduced) process model, containing the information needed to be able to add security-oriented elements. The backward transformation takes into account the initial process and the security enhanced process. In this way, information contained in the initial model lost during forward transformation is recovered. This transformation completes the initial process with the security-oriented elements from the enhanced process model. Elements which have been added to the RCPM process model will be added in the GPPDL model.

Tool-support: Using the proposed metamodels and the Eclipse Modeling Framework, ongoing experimental work is done with SEMCOMDT as a MDE tool-chain supporting the proposed approach metamodels. We build a set of software tools, for designing process models, for populating and for retrieval from the repository. Moreover, we provide tools to support the management of the repository, the generation of documentation and the transformations for refinement and analysis. We choose to derive a text-based syntax to create instances of the metamodel using the Xtext Framework. For the description of the model transformations, the QVT Operational language is used.

VI. Application to a SmartMeter Gateway Case Study

In the context of the TERESA project, we evaluated the approach to describe the development process of a Smart Meter Gateway taking into account the Common Criteria PP for the Gateway of a smart metering system.

Formalized Process: The process consists of nine main phases from Requirement Analysis and System Design, Architecture Design down to System Test and Certification. For demonstration purposes on formalizing this (partially)

security-oriented process model, we will focus on the Architecture Design Phase. In Listing 1 are shown excerpts from the process model.

```
Phase PhSystemArchitectureDesign {
  description 'Defining the Architecture of the
       system-under-development'
  activities {AcSoftwareArchitectureDesign,
       SecArchSoftwareArchitectureDesign,
       AcHardwareArchitectureDesign,
       AcArchitectureSFRTracing} }
```

Listing 1. Extract from Metrology Process Model

Security Process Model Library: In the context, of the SEMCO Project we defined a set of model libraries for different security approaches. Here, we chose to focus on CLASP as an approach giving tool support and being able to be adapted to existing processes.

Adding supplementary Security Concepts to Metrology Process Model: By typing `PhSystemArchitectureDesign` from Listing 1 with `Type_CLASPArchitecturalDesign` from the CLASP Library and by model transformations (Step 2 in Figure 1), we will type existing and obtain new elements in the process model. These elements represent the concepts of CLASP's Architectural Design Phase, net present in the initial process model. An excerpt of CLASP-enhanced PhSystemArchitectureDesign is given in Listing 2. Elements added by the refinement with the CLASP Library are marked with a dashed underline.

```
Phase PhSystemArchitectureDesign {
  description 'Defining the Architecture of the
       system-under-development'
  activities {AcSoftwareArchitectureDesign,
       AcHardwareArchitectureDesign,
       AcArchitectureSFRTracing,
       CLASPSecRiskAssessment3rdParties,
       CLASPSecRequirementsAudit,
       CLASPArchitecture-LevelThreatModeling}
  type Type_CLASPArchitecturalDesign }
SecurityEngineer CLASPSecurityArchitect {
  type Type_CLASPSecurityArchitect }
```

Listing 2. Extract from CLASP-enhanced Metrology Process Model (focus on Architecture Design Phase)

VII. Conclusion

In this paper we propose a methodology to enhance existing process models with security aspect. This is realized by model transformations towards a security-oriented process modeling language RCPM. The security aspects of the modeling language are detailed and demonstrated on a working example through a text-based concrete syntax. The methodology and the modeling language are validated by a use case from the metrology domain. The advantages of the approach are a more direct and intuitive way of adding security concepts to process models for domain having strong security requirements. In addition, assistance is given to the process designer by model type libraries, guiding the designer to conform with security engineering standards, guidelines and/or best practices. Despite the advantages of the approach and RCPM, there are limits to the approach. RCPM is not able to represent all of the concepts given in SPEM2 or other GPPDLs, although this might not raise an issue, since the process concepts needed for security engineering are kept.

Future work fill focus on other security oriented approaches like SecSDM or TSP Secure, which are not yet fully adopted in industry, often lacking support for modeling processes with their support. Another point for further improvements is additional validation with industry-based and standard process models, pushing conformance validation towards security engineering standards and or guidelines, as well as extending usability testing.

References

[1] B. Selic, "The pragmatics of model-driven development," *IEEE Softw.*, vol. 20, no. 5, pp. 19–25, 2003.

[2] C. Gonzalez-Perez and B. Henderson-Sellers, "Modelling software development methodologies: A conceptual foundation," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1778–1796, Nov. 2007.

[3] OMG, "Software & systems process engineering metamodel specification (SPEM) version 2.0," Object Management Group, Inc., Tech. Rep., 2008.

[4] P. Kruchten, *The Rational Unified Process: An Introduction*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[5] C. Hug, A. Front, D. Rieu, and B. Henderson-Sellers, "A method to build information systems engineering process metamodels," *Journal of Systems and Software*, vol. 82, no. 10, pp. 1730–1742, Oct. 2009.

[6] K. Forsberg, H. Mooz, and H. Cotterman, *Visualizing Project Management, A Model for Business and Technical Success*, 2nd ed. Wiley, 2000.

[7] Microsoft, "Microsoft security development lifecycle (SDL) process guidance - version 5.2," 2012.

[8] OWASP, *OWASP CLASP V1.2*. OWASP, Nov. 2007.

[9] G. McGraw, *Software security: building security in*. Addison-Wesley Professional, 2006.

[10] Common Criteria, "Common criteria for information technology security evaluation v3.1r4," Common Criteria, Tech. Rep. CCMB-2012-09-001/002/003, 2012.

[11] J. Geisel, B. Hamid, and J.-M. Bruel, "Repository-centric process modeling – example of a pattern based development process," in *Software Engineering Research, Management and Applications*. Springer International Publishing, 2014, no. 496, pp. 247–261.

[12] B. Hamid, J. Geisel, A. Ziani, and D. Gonzalez, "Safety lifecycle development process modeling for embedded systems - example of railway domain," in *Software Engineering for Resilient Systems*. Springer Berlin Heidelberg, 2012, no. 7527, pp. 63–75.